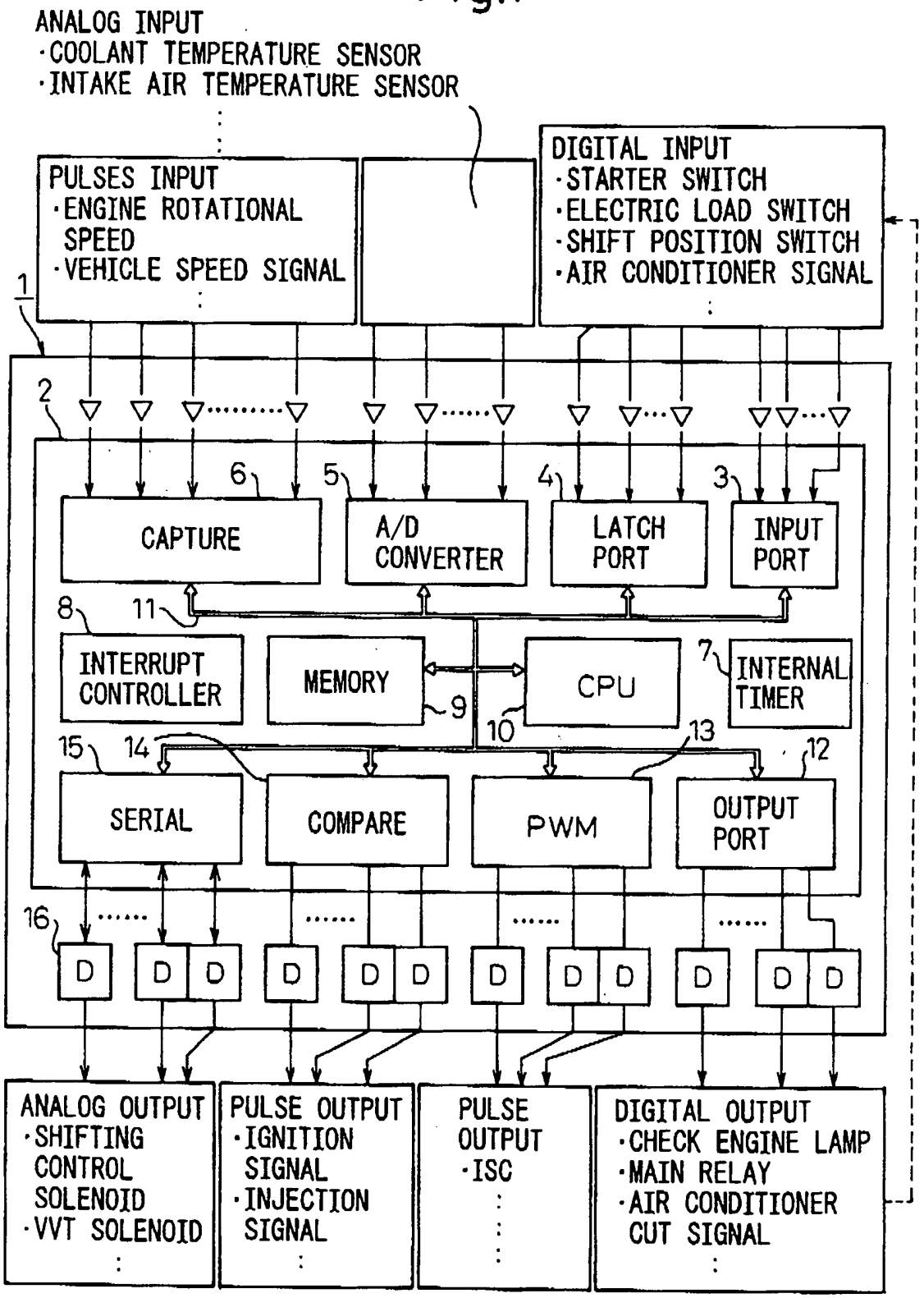
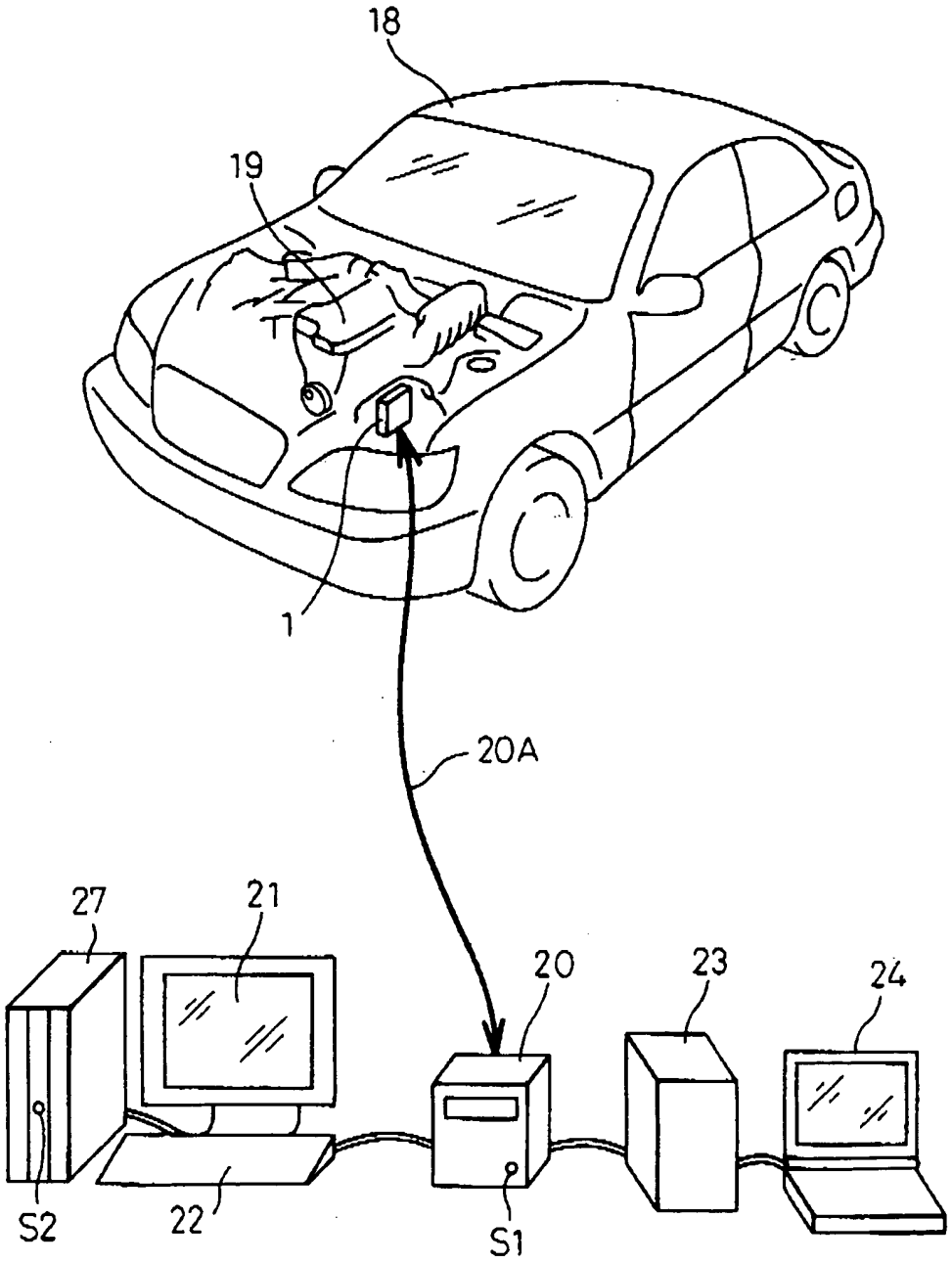
1/25

Fig.1


ANALOG INPUT
·COOLANT TEMPERATURE SENSOR
·INTAKE AIR TEMPERATURE SENSOR
PULSES INPUT
·ENGINE ROTATIONAL SPEED
·VEHICLE SPEED SIGNAL
DIGITAL INPUT
·STARTER SWITCH
·ELECTRIC LOAD SWITCH
·SHIFT POSITION SWITCH
·AIR CONDITIONER SIGNAL
1
2
CAPTURE
A/D CONVERTER
LATCH PORT
INPUT PORT
6
5
4
3
8
11
INTERRUPT CONTROLLER
MEMORY
CPU
INTERNAL TIMER
9
10
7
15
14
13
12
SERIAL
COMPARE
PWM
OUTPUT PORT
16
D
ANALOG OUTPUT
·SHIFTING CONTROL SOLENOID
·VVT SOLENOID
PULSE OUTPUT
·IGNITION SIGNAL
·INJECTION SIGNAL
PULSE OUTPUT
·ISC
DIGITAL OUTPUT
·CHECK ENGINE LAMP
·MAIN RELAY
·AIR CONDITIONER CUT SIGNAL

2/25

Fig.2



3/25

Fig.3

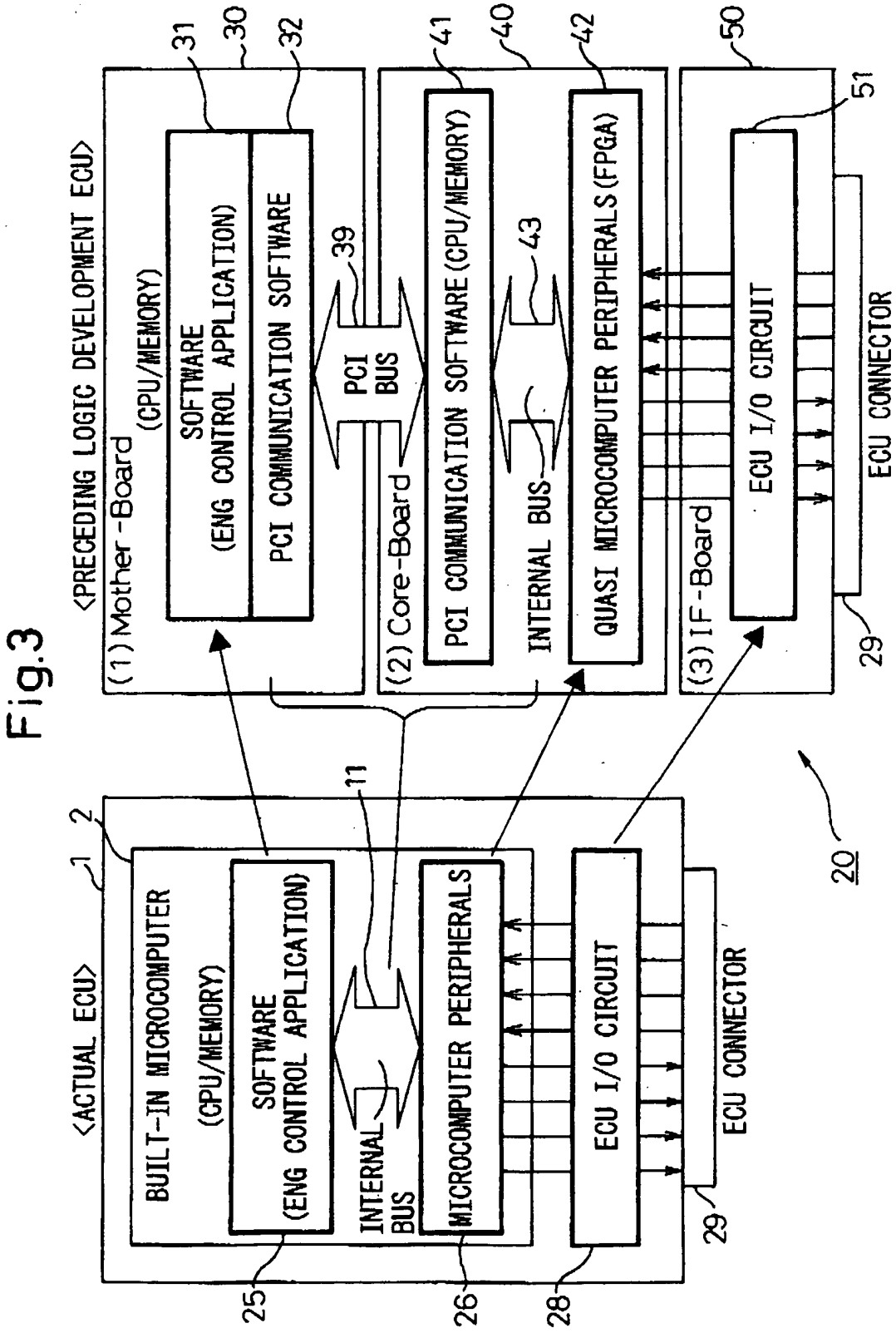
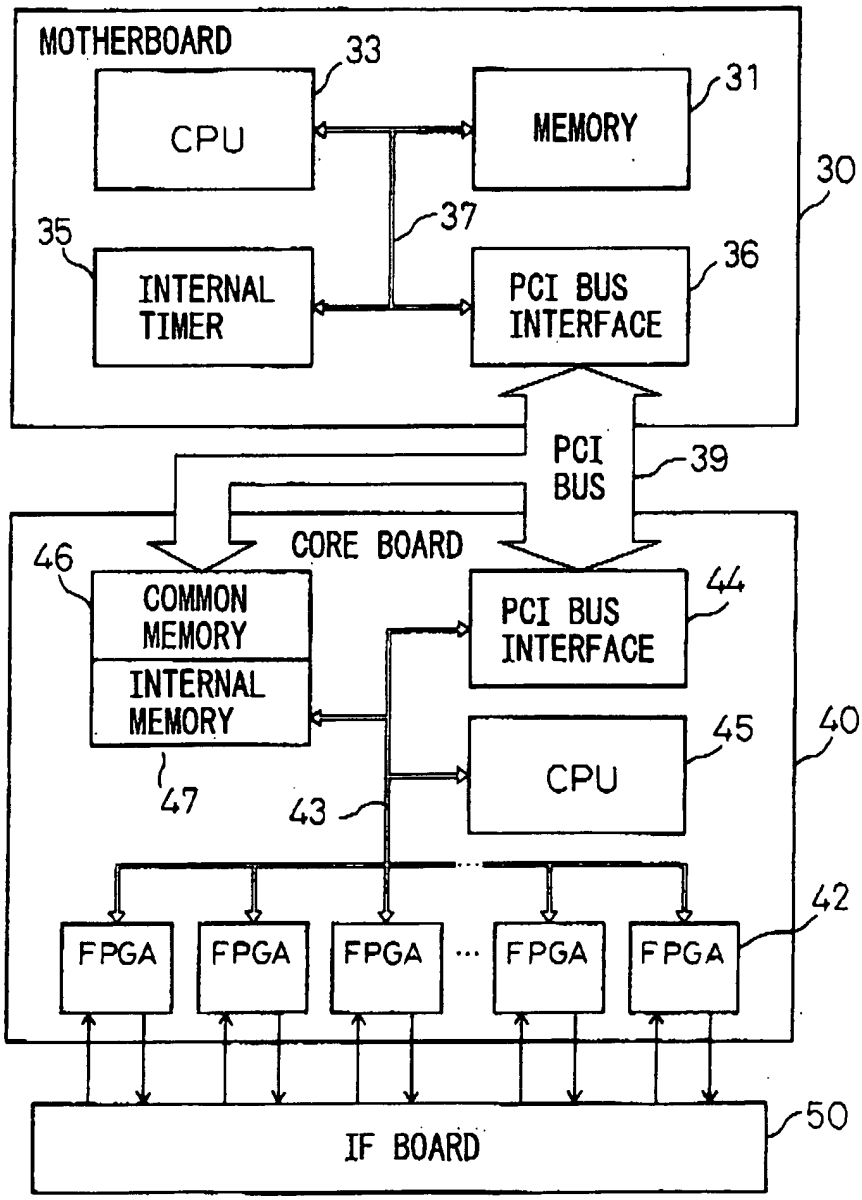
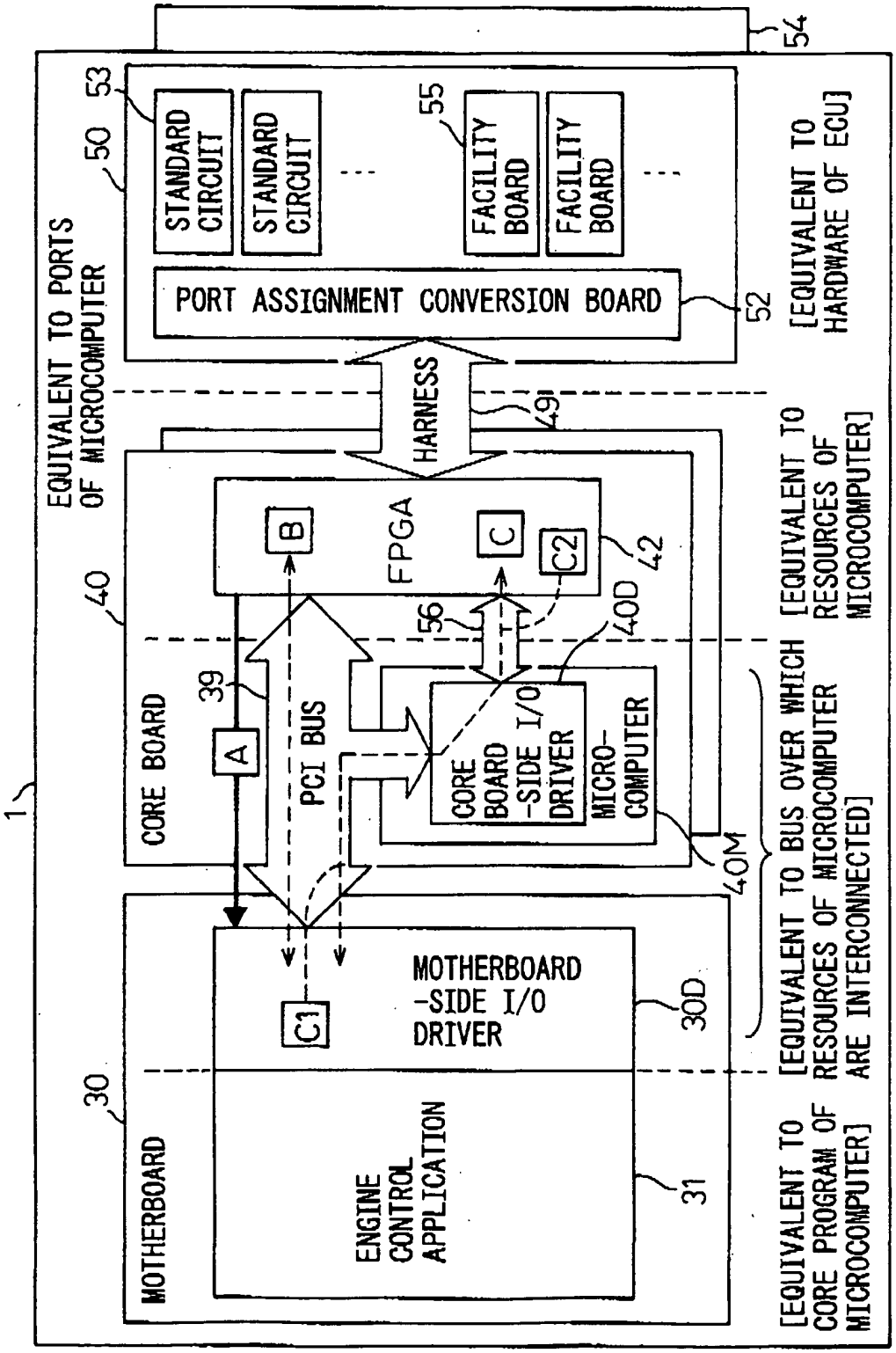

<PRECEDING LOGIC DEVELOPMENT ECU>
(1) Mother-Board
(CPU/MEMORY)
SOFTWARE
(ENG CONTROL APPLICATION)
PCI COMMUNICATION SOFTWARE
PCI BUS
(2) Core-Board
PCI COMMUNICATION SOFTWARE (CPU/MEMORY)
INTERNAL BUS
QUASI MICROCOMPUTER PERIPHERALS (FPGA)
(3) IF-Board
ECU I/O CIRCUIT
ECU CONNECTOR
<ACTUAL ECU>
BUILT-IN MICROCOMPUTER
(CPU/MEMORY)
SOFTWARE
(ENG CONTROL APPLICATION)
INTERNAL BUS
MICROCOMPUTER PERIPHERALS
ECU I/O CIRCUIT
ECU CONNECTOR
20
25
26
28
29
30
31
32
39
40
41
42
43
50
51

Fig.4


MOTHERBOARD
33
CPU
MEMORY
31
30
37
35
INTERNAL TIMER
PCI BUS INTERFACE
36
PCI BUS
39
CORE BOARD
46
COMMON MEMORY
INTERNAL MEMORY
47
PCI BUS INTERFACE
44
CPU
45
40
43
FPGA
FPGA
FPGA
FPGA
FPGA
42
IF BOARD
50

5/25

Fig.5


1
54
53
50
STANDARD CIRCUIT
STANDARD CIRCUIT
55
FACILITY BOARD
FACILITY BOARD
PORT ASSIGNMENT CONVERSION BOARD
52
EQUIVALENT TO PORTS OF MICROCOMPUTER
[EQUIVALENT TO HARDWARE OF ECU]
HARNESS
49
40
B
FPGA
C
C2
42
56
40D
39
A
PCI BUS
CORE BOARD -SIDE I/O DRIVER
MICRO-COMPUTER
CORE BOARD
40M
[EQUIVALENT TO RESOURCES OF MICROCOMPUTER]
[EQUIVALENT TO BUS OVER WHICH RESOURCES OF MICROCOMPUTER ARE INTERCONNECTED]
C1
MOTHERBOARD -SIDE I/O DRIVER
30D
30
MOTHERBOARD
ENGINE CONTROL APPLICATION
31
[EQUIVALENT TO CORE PROGRAM OF MICROCOMPUTER]

6/25

Fig.6

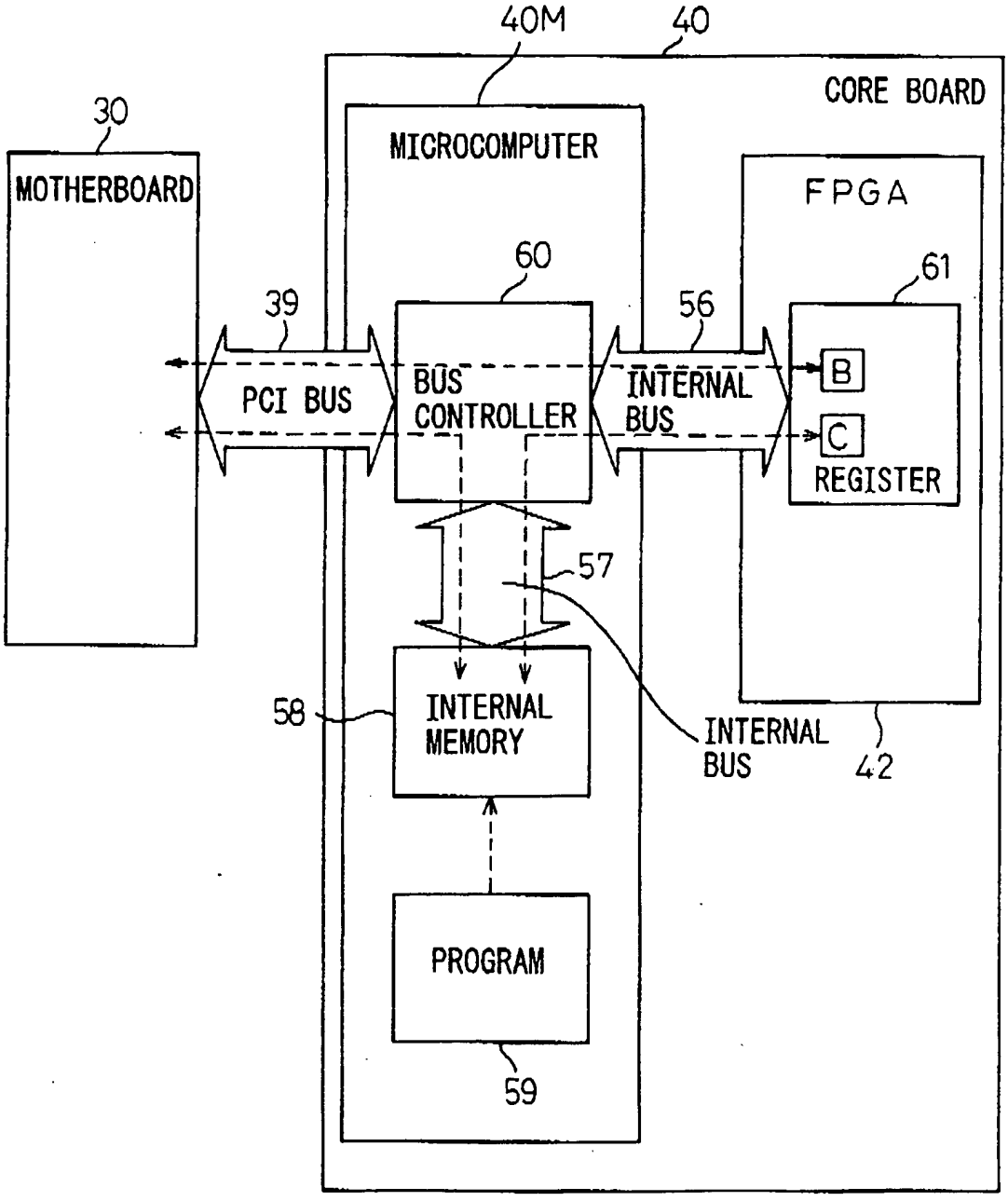

40M
40
CORE BOARD
30
MOTHERBOARD
MICROCOMPUTER
FPGA
60
61
39
56
PCI BUS
BUS CONTROLLER
INTERNAL BUS
B
C
REGISTER
57
58
INTERNAL MEMORY
INTERNAL BUS
42
PROGRAM
59

Fig.7A

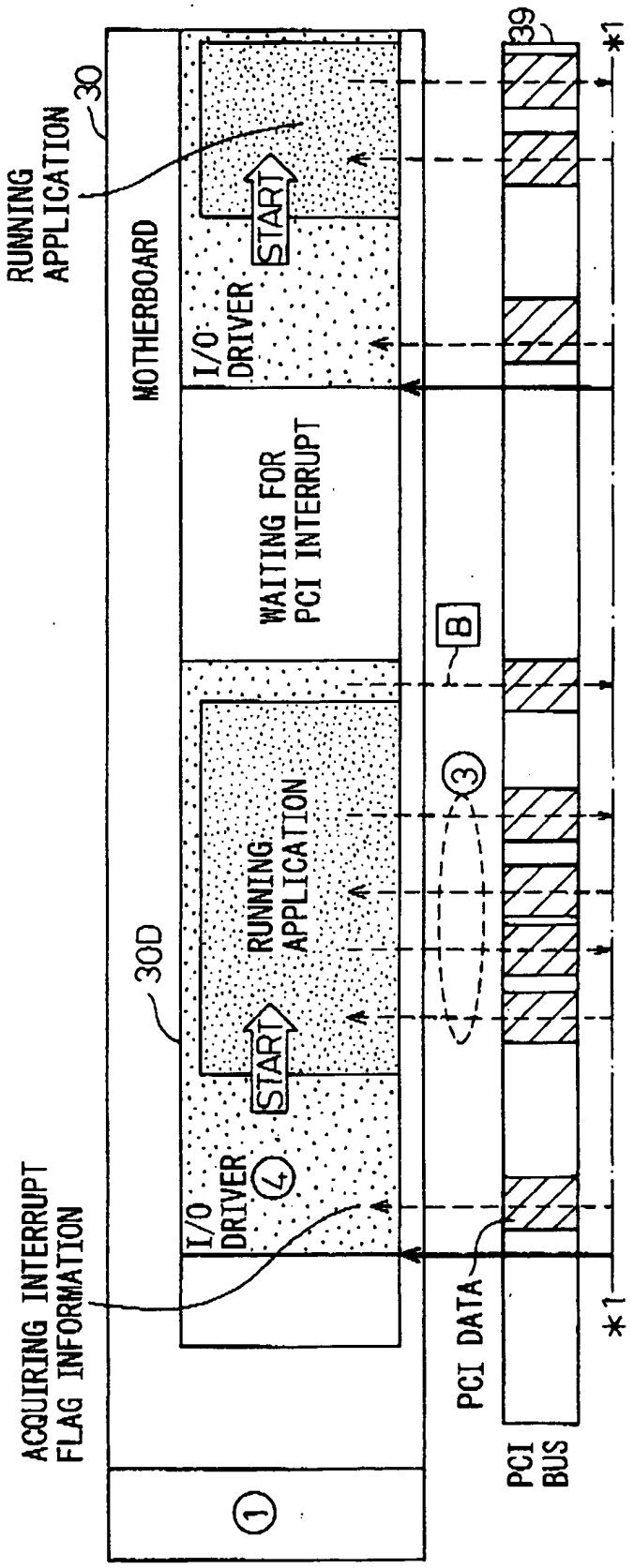
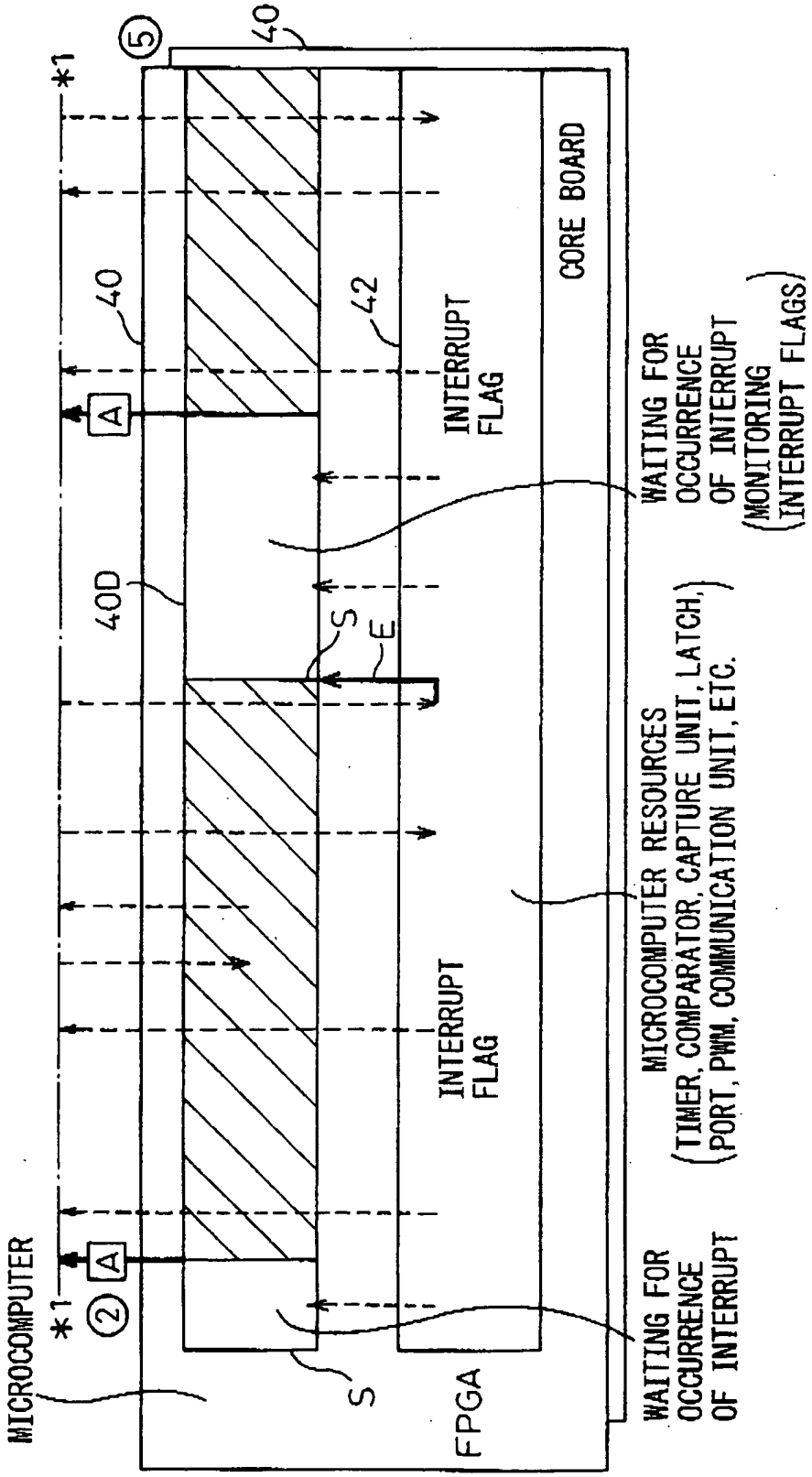

RUNNING APPLICATION
30
MOTHERBOARD
I/O DRIVER
START
WAITING FOR PCI INTERRUPT
B
30D
RUNNING APPLICATION
START
I/O DRIVER
4
3
ACQUIRING INTERRUPT FLAG INFORMATION
1
PCI DATA
39
*1
*1
PCI BUS

Fig.7B


MICROCOMPUTER
*1
A
2
*1
A
40
5
40
42
40D
S
E
S
INTERRUPT FLAG
INTERRUPT FLAG
CORE BOARD
FPGA
WAITING FOR OCCURRENCE OF INTERRUPT
MICROCOMPUTER RESOURCES (TIMER, COMPARATOR, CAPTURE UNIT, LATCH, PORT, PWM, COMMUNICATION UNIT, ETC.)
WAITING FOR OCCURRENCE OF INTERRUPT (MONITORING INTERRUPT FLAGS)

9/25

Fig.8A

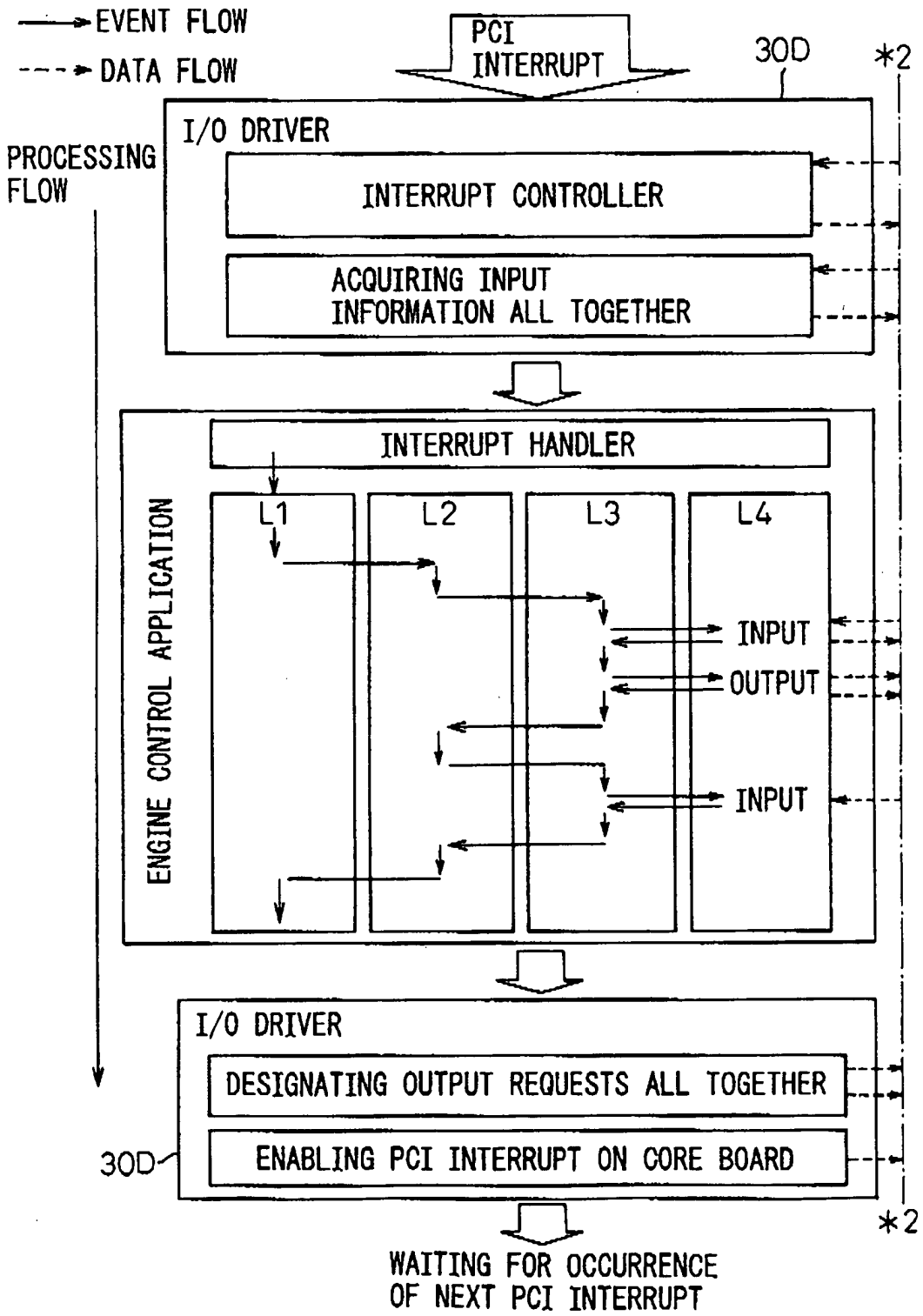

EVENT FLOW
DATA FLOW
PCI INTERRUPT
30D
*2
PROCESSING FLOW
I/O DRIVER
INTERRUPT CONTROLLER
ACQUIRING INPUT INFORMATION ALL TOGETHER
INTERRUPT HANDLER
ENGINE CONTROL APPLICATION
L1
L2
L3
L4
INPUT
OUTPUT
INPUT
I/O DRIVER
DESIGNATING OUTPUT REQUESTS ALL TOGETHER
ENABLING PCI INTERRUPT ON CORE BOARD
30D
*2
WAITING FOR OCCURRENCE OF NEXT PCI INTERRUPT

Fig.8B

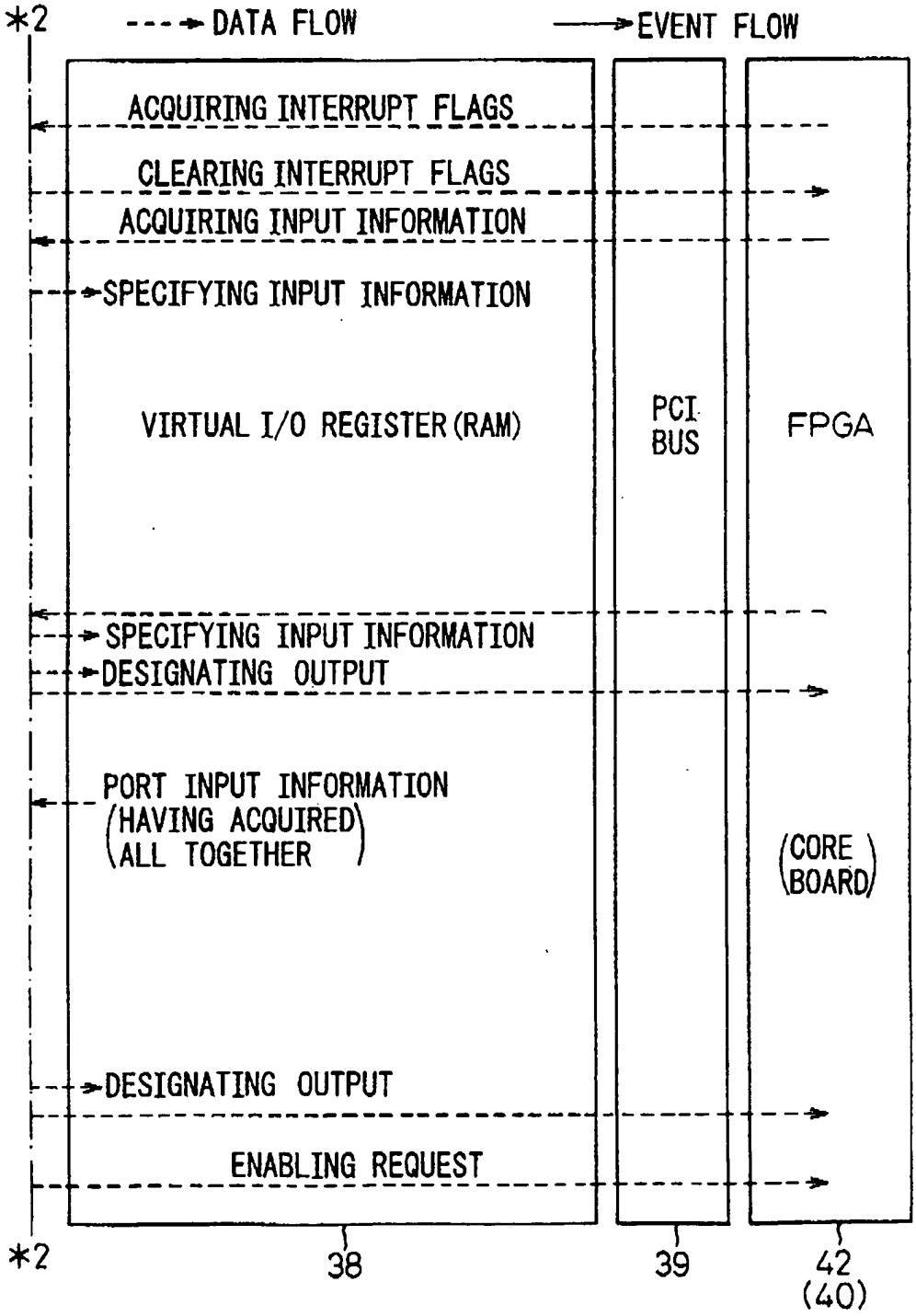

*2
DATA FLOW
EVENT FLOW
ACQUIRING INTERRUPT FLAGS
CLEARING INTERRUPT FLAGS
ACQUIRING INPUT INFORMATION
SPECIFYING INPUT INFORMATION
VIRTUAL I/O REGISTER (RAM)
PCI BUS
FPGA
SPECIFYING INPUT INFORMATION
DESIGNATING OUTPUT
PORT INPUT INFORMATION (HAVING ACQUIRED ALL TOGETHER)
(CORE BOARD)
DESIGNATING OUTPUT
ENABLING REQUEST
*2
38
39
42 (40)

Fig.9

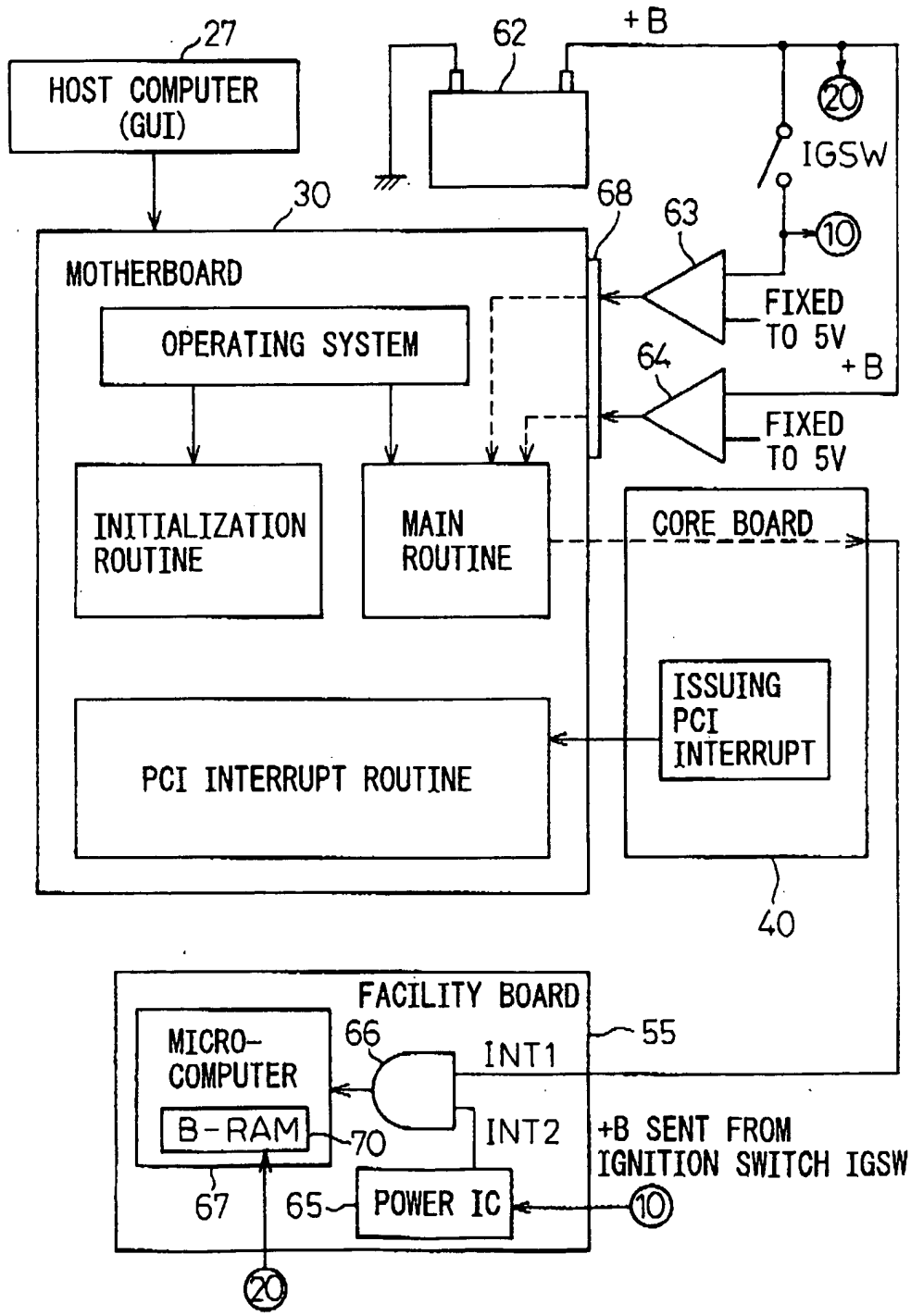

27
HOST COMPUTER (GUI)
62
+B
20
IGSW
30
68
63
10
MOTHERBOARD
OPERATING SYSTEM
FIXED TO 5V
+B
64
FIXED TO 5V
INITIALIZATION ROUTINE
MAIN ROUTINE
CORE BOARD
ISSUING PCI INTERRUPT
PCI INTERRUPT ROUTINE
40
FACILITY BOARD
MICRO-COMPUTER
66
INT1
55
B-RAM
70
INT2
+B SENT FROM IGNITION SWITCH IGSW
67
65
POWER IC
10
20

Fig.10A

	MOTHERBOARD	CORE BOARD	FACILITY BOARD
1	AN ENGINEER TURNS ON THE POWER SUPPLY OF THE SYSTEM.	DITTO	THE POWER SUPPLY IS OFF.
2	SOFTWARE IS DOWNLOADED FROM THE HOST COMPUTER, AND THE MOTHERBOARD IS ACTIVATED.	DITTO PCI INTERRUPTS ARE LOCKED.	
3	INITIAL VALUES TO WHICH PORTS ARE SET ARE DETERMINED. (※1) (1) THE COMPARATOR IS SET TO AN IMMEDIATE OUTPUT MODE. (2) THE PWM IS SET TO A 0% OUTPUT MODE. (3) THE SIGNAL INIT1 IS DRIVEN LOW. (4) THE MAIN ROUTINE IS SET TO STATE 1.		
4	ACTION OF STATE 1 (1) A WAIT STATE IS MAINTAINED UNTIL THE SWITCH IGSW IS TURNED ON.		
5	(2) WHETHER THE SWITCH IGSW IS TURNED ON IS DETECTED. (3) THE MAIN ROUTINE IS SET TO STATE 2.		THE POWER SUPPLY IS TURNED ON.

Fig.10B

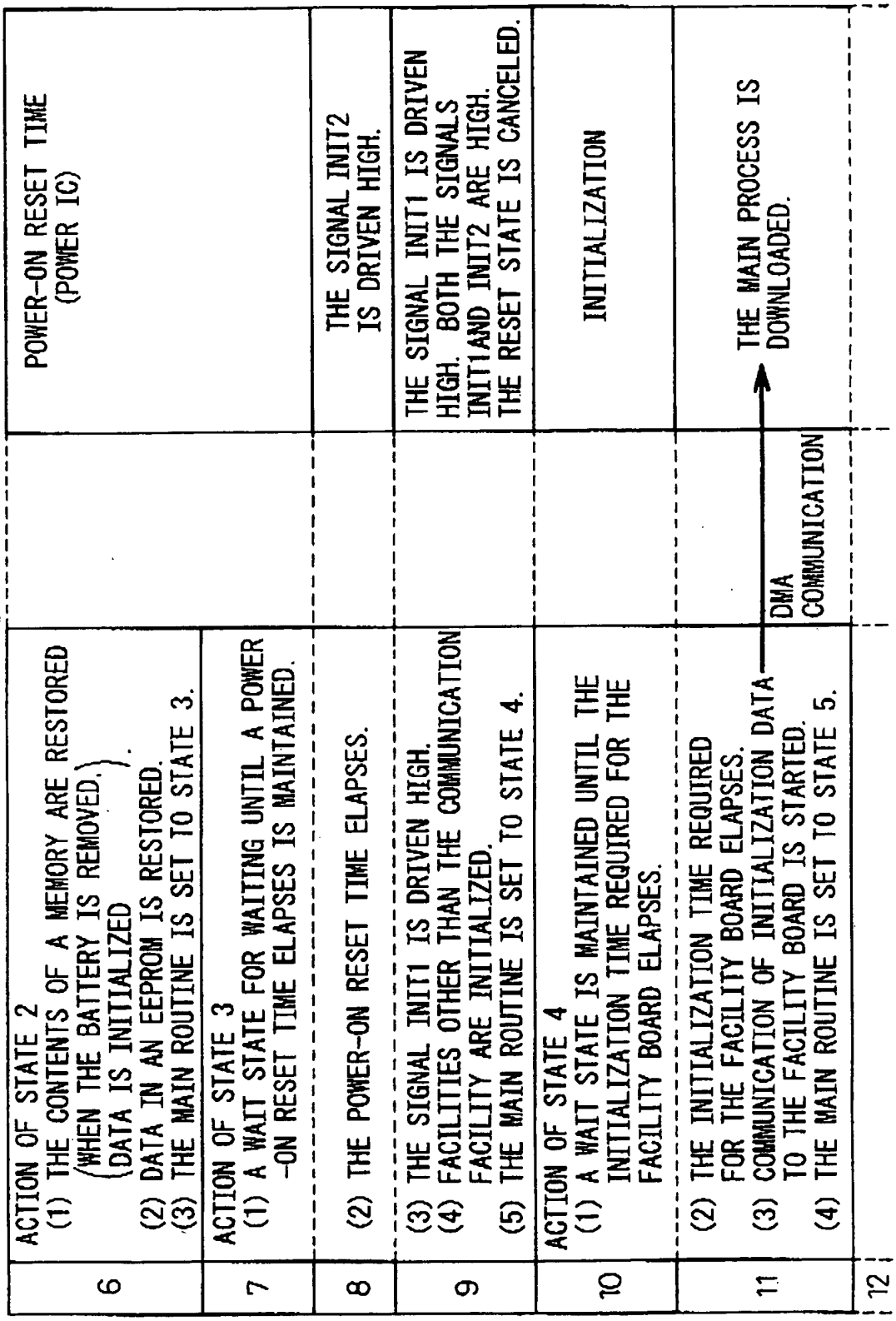

6
ACTION OF STATE 2
(1) THE CONTENTS OF A MEMORY ARE RESTORED (WHEN THE BATTERY IS REMOVED.) (DATA IS INITIALIZED).
(2) DATA IN AN EEPROM IS RESTORED.
(3) THE MAIN ROUTINE IS SET TO STATE 3.
POWER-ON RESET TIME (POWER IC)
7
ACTION OF STATE 3
(1) A WAIT STATE FOR WAITING UNTIL A POWER -ON RESET TIME ELAPSES IS MAINTAINED.
8
(2) THE POWER-ON RESET TIME ELAPSES.
THE SIGNAL INIT2 IS DRIVEN HIGH.
9
(3) THE SIGNAL INIT1 IS DRIVEN HIGH.
(4) FACILITIES OTHER THAN THE COMMUNICATION FACILITY ARE INITIALIZED.
(5) THE MAIN ROUTINE IS SET TO STATE 4.
THE SIGNAL INIT1 IS DRIVEN HIGH. BOTH THE SIGNALS INIT1AND INIT2 ARE HIGH.
THE RESET STATE IS CANCELED.
10
ACTION OF STATE 4
(1) A WAIT STATE IS MAINTAINED UNTIL THE INITIALIZATION TIME REQUIRED FOR THE FACILITY BOARD ELAPSES.
INITIALIZATION
11
(2) THE INITIALIZATION TIME REQUIRED FOR THE FACILITY BOARD ELAPSES.
(3) COMMUNICATION OF INITIALIZATION DATA TO THE FACILITY BOARD IS STARTED.
(4) THE MAIN ROUTINE IS SET TO STATE 5.
DMA COMMUNICATION
THE MAIN PROCESS IS DOWNLOADED.
12

Fig.11A

	MOTHERBOARD	CORE BOARD	FACILITY BOARD
12	ACTION OF STATE 5 (1) A WAIT STATE IS MAINTAINED UNTIL COMMUNICATION OF INITIALIZATION DATA TO THE FACILITY BOARD IS COMPLETED.		DATA IS RECEIVED, AND TRANSMISSION DATA IS PRODUCED.
13	(2) SENSING COMPLETION ※IF A TIMEOUT OCCURS WITHOUT COMPLETION, THE MAIN ROUTINE IS RETURNED TO STATE 4.		THE TRANSMISSION DATA IS TRANSMITTED.
14	ACTION OF STATE 6 (1) INITIALIZATION IS COMPLETED.		
15	(2) A REQUEST FOR UNLOCKING PCI INTERRUPTS IS ISSUED TO THE CORE BOARD. (3) THE MAIN ROUTINE IS SET TO STATE 7.	ALL THE INTERRUPT FLAGS ARE CLEARED. PCI INTERRUPTS ARE UNLOCKED.	

Fig.11B

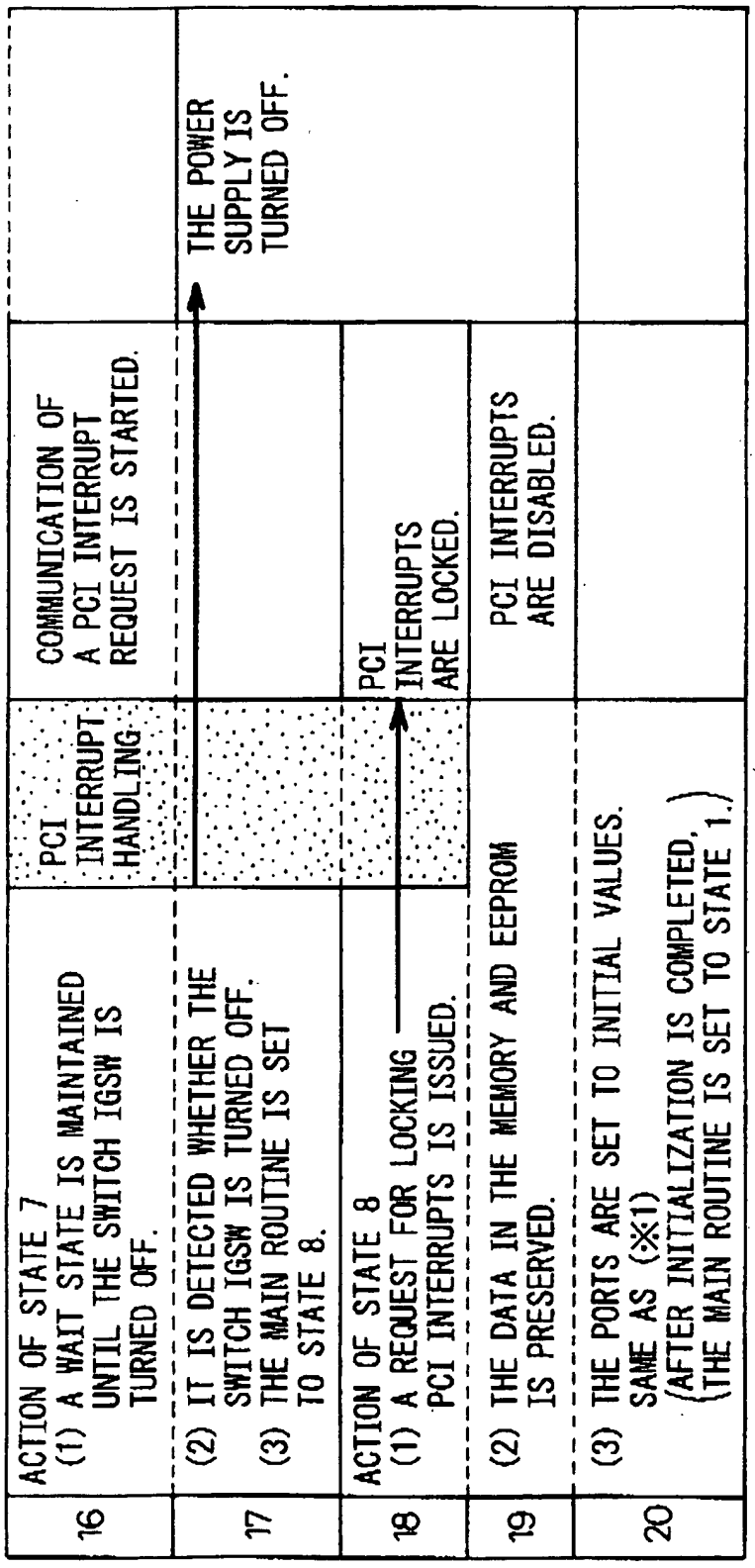

16
ACTION OF STATE 7
(1) A WAIT STATE IS MAINTAINED UNTIL THE SWITCH IGSW IS TURNED OFF.
PCI INTERRUPT HANDLING
COMMUNICATION OF A PCI INTERRUPT REQUEST IS STARTED.
17
(2) IT IS DETECTED WHETHER THE SWITCH IGSW IS TURNED OFF.
(3) THE MAIN ROUTINE IS SET TO STATE 8.
THE POWER SUPPLY IS TURNED OFF.
18
ACTION OF STATE 8
(1) A REQUEST FOR LOCKING PCI INTERRUPTS IS ISSUED.
PCI INTERRUPTS ARE LOCKED.
19
(2) THE DATA IN THE MEMORY AND EEPROM IS PRESERVED.
PCI INTERRUPTS ARE DISABLED.
20
(3) THE PORTS ARE SET TO INITIAL VALUES. SAME AS (※1)
(AFTER INITIALIZATION IS COMPLETED, THE MAIN ROUTINE IS SET TO STATE 1.)

Fig.12A

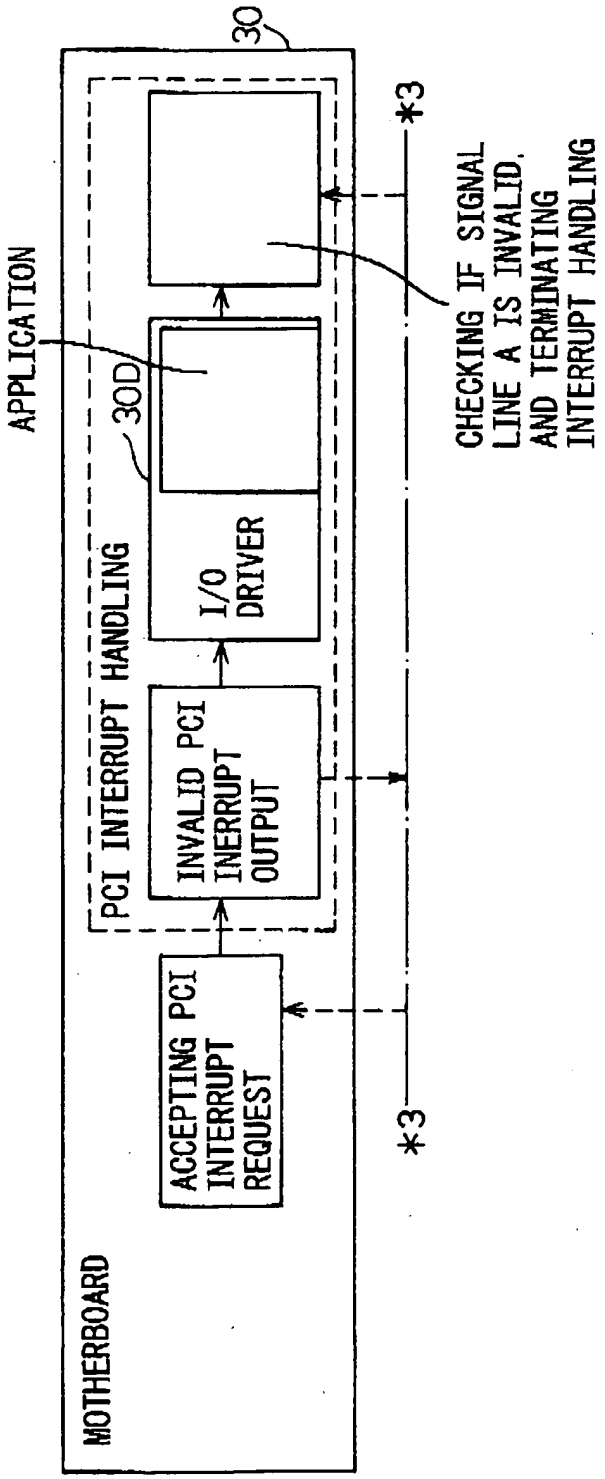

30
*3
APPLICATION
30D
I/O DRIVER
PCI INTERRUPT HANDLING
INVALID PCI INERRUPT OUTPUT
ACCEPTING PCI INTERRUPT REQUEST
MOTHERBOARD
*3
CHECKING IF SIGNAL LINE A IS INVALID, AND TERMINATING INTERRUPT HANDLING

Fig.12B

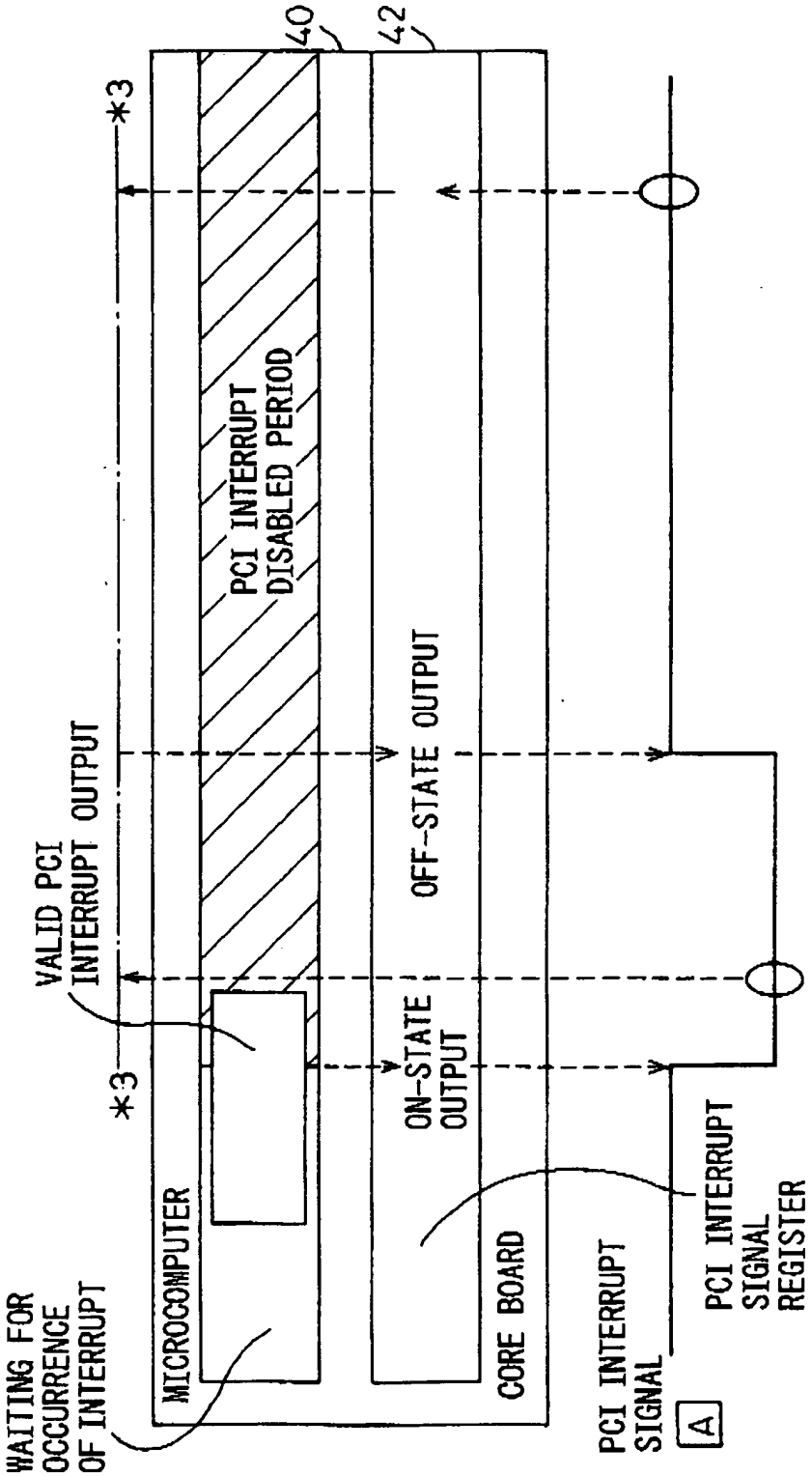

40
42
*3
*3
PCI INTERRUPT DISABLED PERIOD
VALID PCI INTERRUPT OUTPUT
OFF-STATE OUTPUT
ON-STATE OUTPUT
MICROCOMPUTER
CORE BOARD
WAITING FOR OCCURRENCE OF INTERRUPT
PCI INTERRUPT SIGNAL
A
PCI INTERRUPT SIGNAL REGISTER

Fig.13A

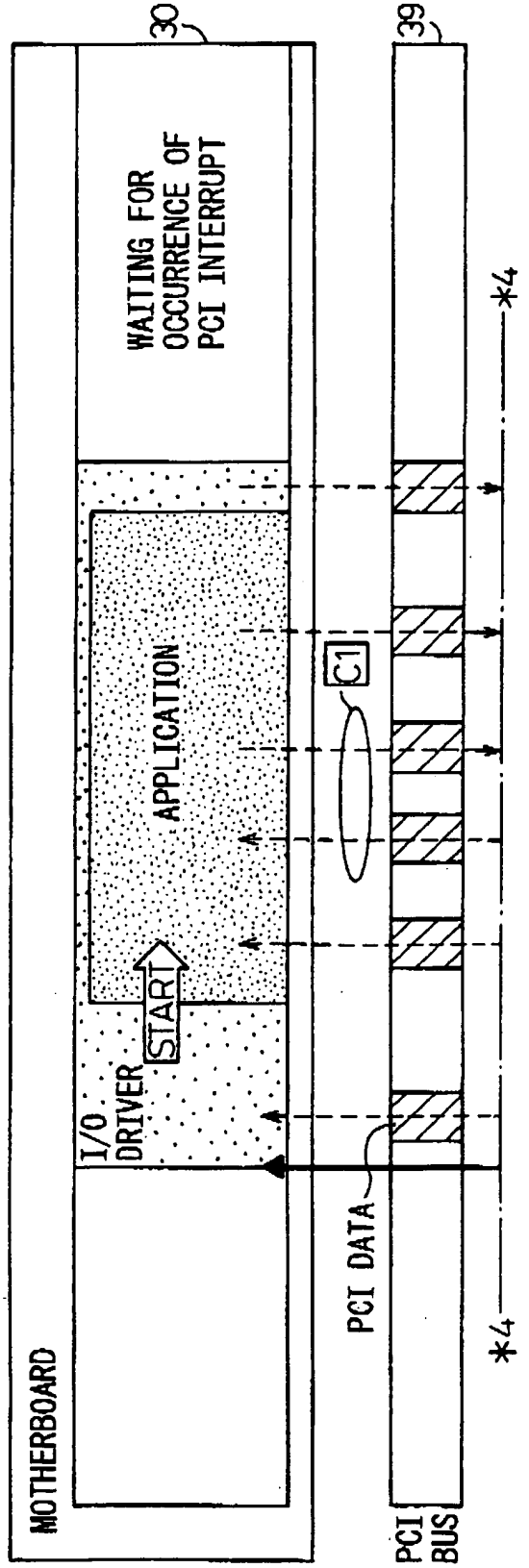

30
39
WAITING FOR OCCURRENCE OF PCI INTERRUPT
APPLICATION
C1
START
I/O DRIVER
PCI DATA
*4
*4
MOTHERBOARD
PCI BUS

Fig.13B

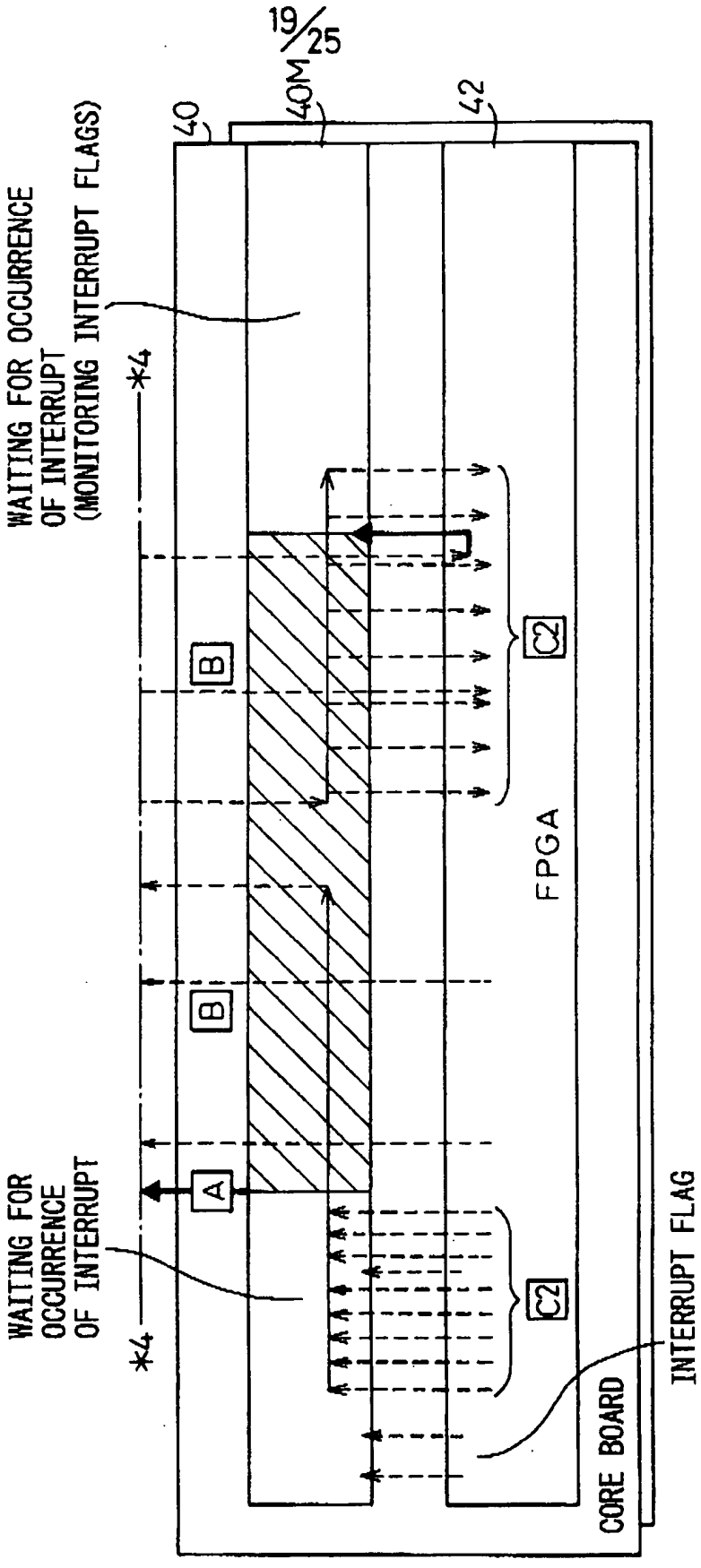

40
40M
42
WAITING FOR OCCURRENCE OF INTERRUPT (MONITORING INTERRUPT FLAGS)
*4
B
B
A
C2
C2
FPGA
WAITING FOR OCCURRENCE OF INTERRUPT
*4
CORE BOARD
INTERRUPT FLAG

Fig.14

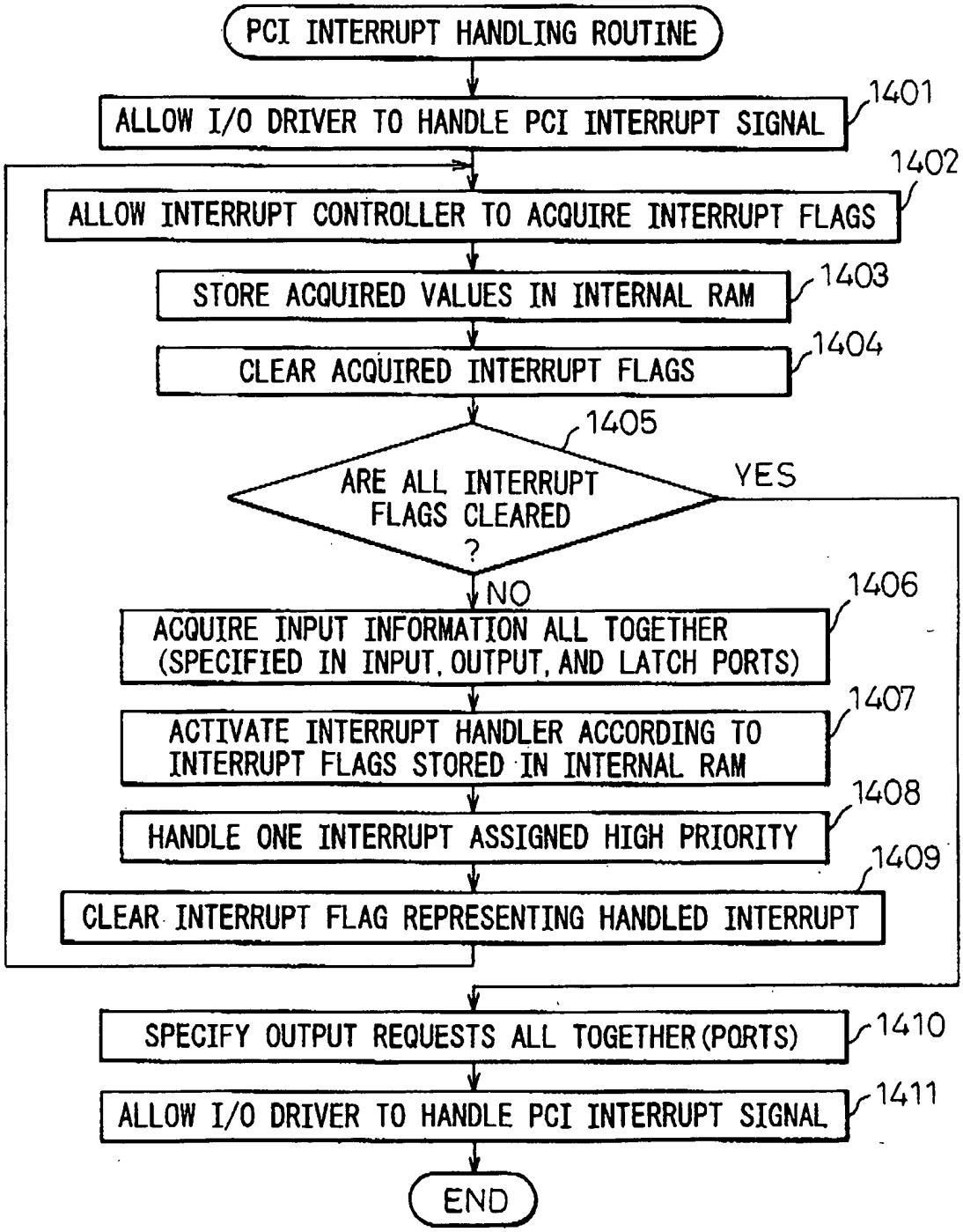

PCI INTERRUPT HANDLING ROUTINE
ALLOW I/O DRIVER TO HANDLE PCI INTERRUPT SIGNAL
1401
ALLOW INTERRUPT CONTROLLER TO ACQUIRE INTERRUPT FLAGS
1402
STORE ACQUIRED VALUES IN INTERNAL RAM
1403
CLEAR ACQUIRED INTERRUPT FLAGS
1404
ARE ALL INTERRUPT FLAGS CLEARED ?
1405
YES
NO
ACQUIRE INPUT INFORMATION ALL TOGETHER (SPECIFIED IN INPUT, OUTPUT, AND LATCH PORTS)
1406
ACTIVATE INTERRUPT HANDLER ACCORDING TO INTERRUPT FLAGS STORED IN INTERNAL RAM
1407
HANDLE ONE INTERRUPT ASSIGNED HIGH PRIORITY
1408
CLEAR INTERRUPT FLAG REPRESENTING HANDLED INTERRUPT
1409
SPECIFY OUTPUT REQUESTS ALL TOGETHER (PORTS)
1410
ALLOW I/O DRIVER TO HANDLE PCI INTERRUPT SIGNAL
1411
END

Fig.15

(a)

.....	CH5	CH4	CH3	CH2	CH1	CH0
.....	0	0	1	0	0	1

READ 1 AS PRESENCE OF AN INTERRUPT FACTOR,
AND READ 0 AS ABSENCE OF AN INTERRUPT FACTOR.
WRITE 1 TO CLEAR DATA, AND WRITE 0 TO PRESERVE DATA.

(b)

CAPTURED VALUE ON CHANNEL CH0	
CAPTURED VALUE ON CHANNEL CH1	
CAPTURED VALUE ON CHANNEL CH2	
CAPTURED VALUE ON CHANNEL CH3	
CAPTURED VALUE ON CHANNEL CH4	
CAPTURED VALUE ON CHANNEL CH5	

32 BITS IN WIDTH

Fig.16

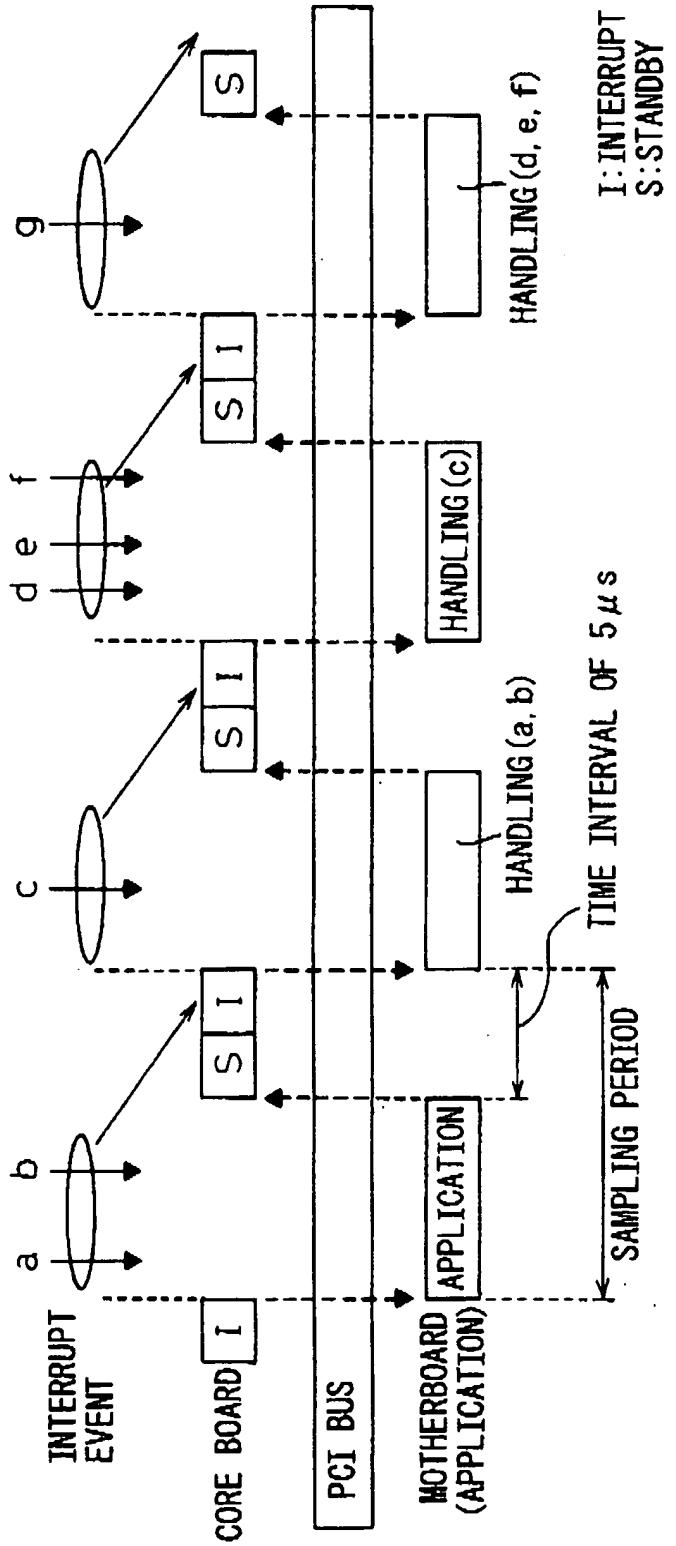

INTERRUPT EVENT
CORE BOARD
PCI BUS
MOTHERBOARD (APPLICATION)
I
S
I
S
I
S
I
S
I
S
a
b
c
d
e
f
g
APPLICATION
HANDLING (a, b)
HANDLING (c)
HANDLING (d, e, f)
TIME INTERVAL OF 5 μs
SAMPLING PERIOD
I: INTERRUPT
S: STANDBY

Fig.17

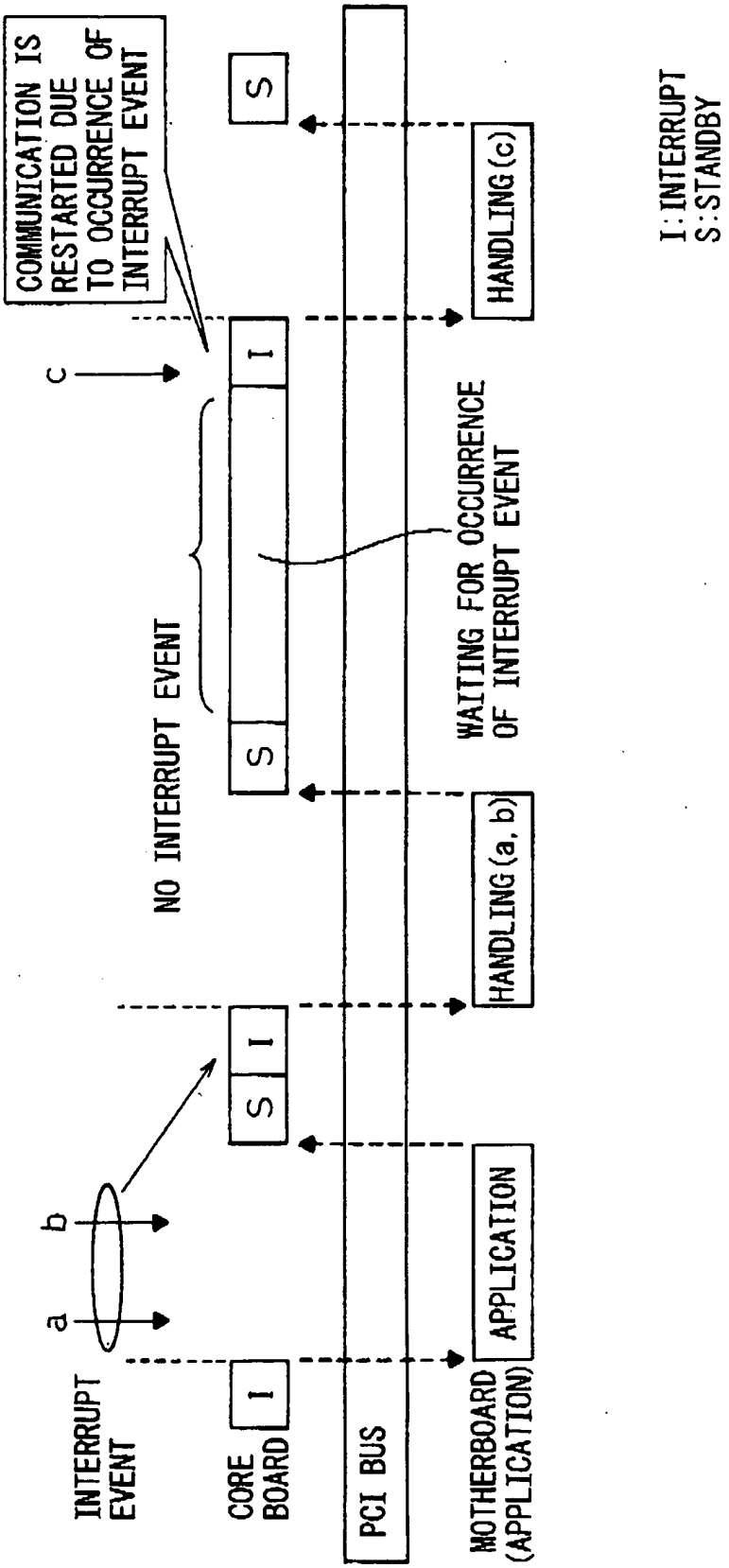

INTERRUPT EVENT
CORE BOARD
PCI BUS
MOTHERBOARD (APPLICATION)
APPLICATION
a
b
c
I
S
HANDLING(a, b)
HANDLING(c)
NO INTERRUPT EVENT
WAITING FOR OCCURRENCE OF INTERRUPT EVENT
COMMUNICATION IS RESTARTED DUE TO OCCURRENCE OF INTERRUPT EVENT
I: INTERRUPT
S: STANDBY

Fig.18A

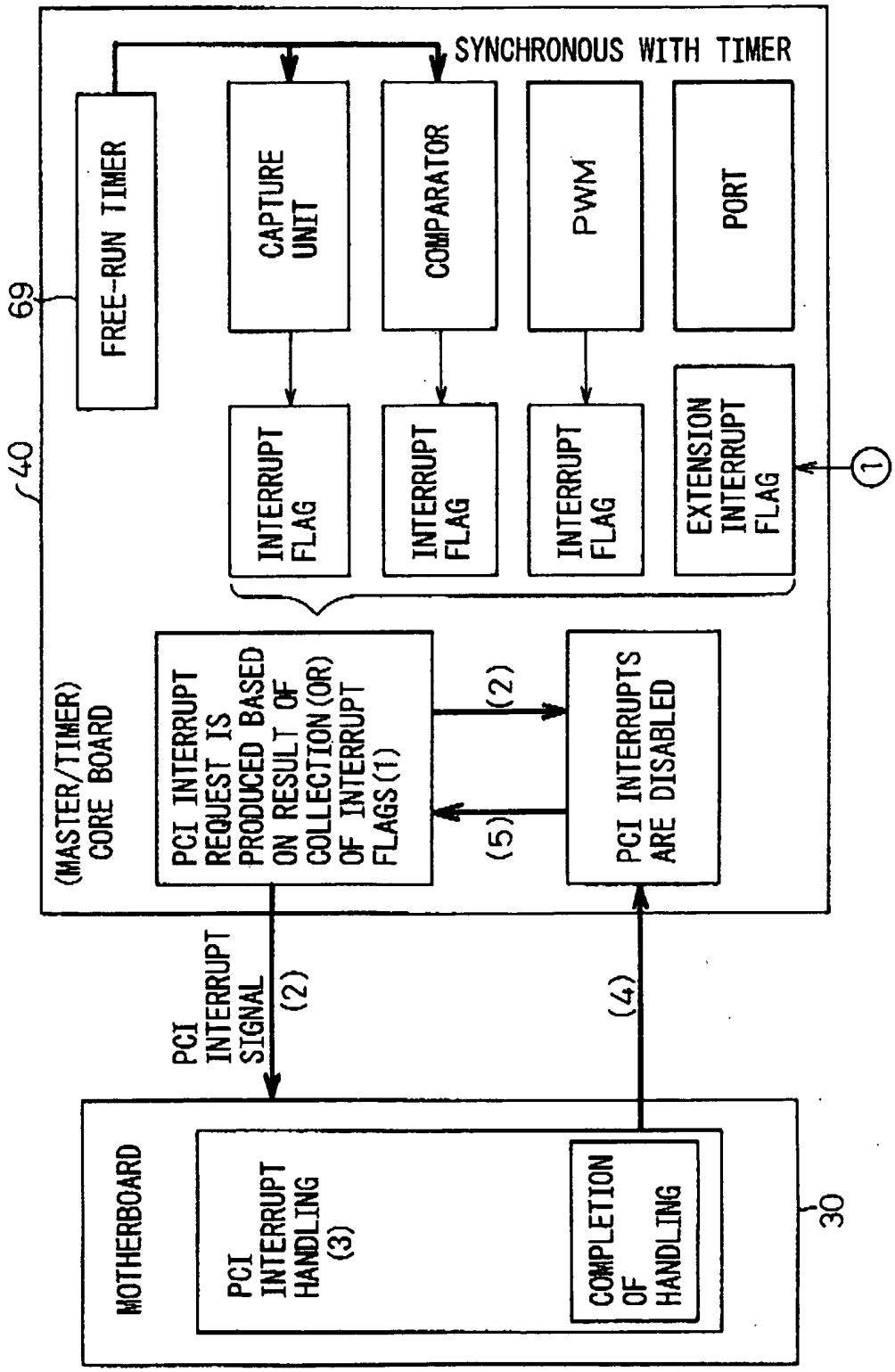
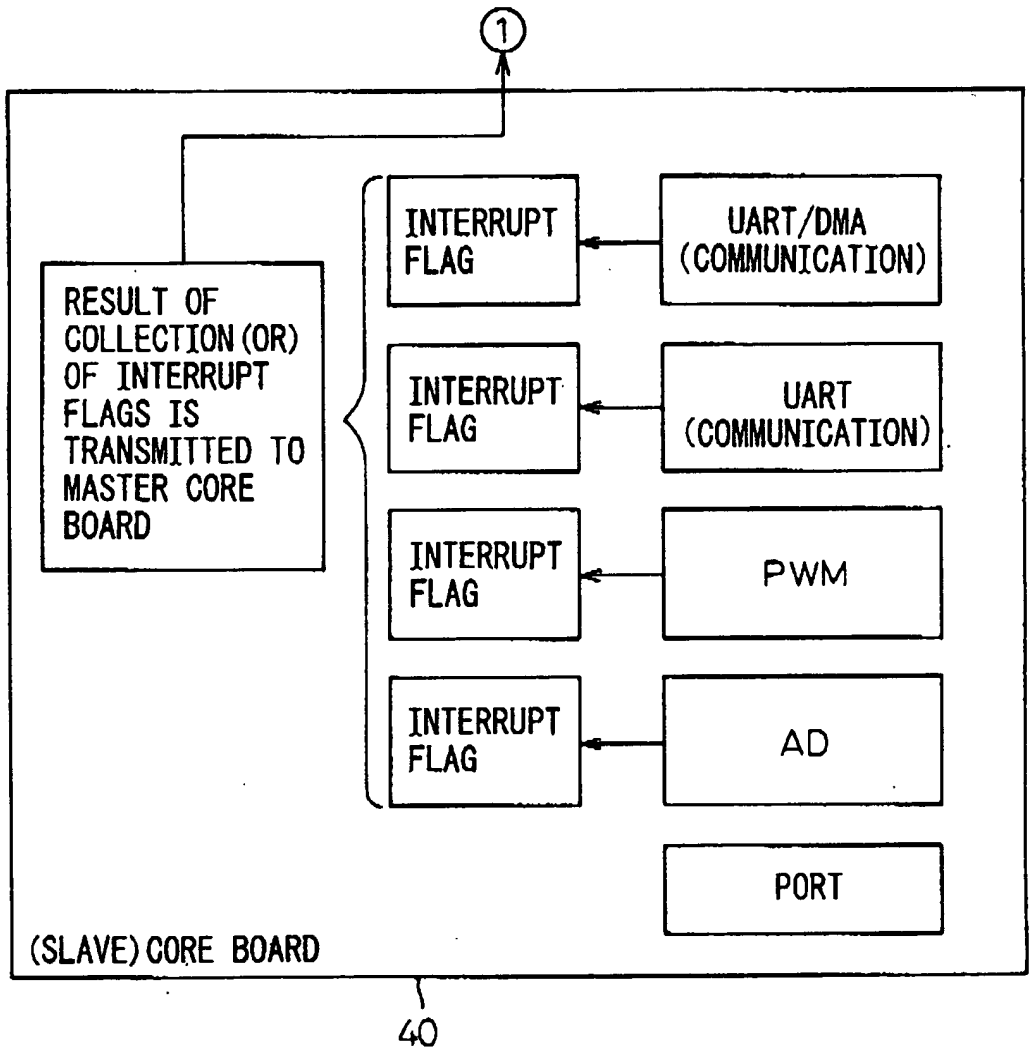

SYNCHRONOUS WITH TIMER
FREE-RUN TIMER
CAPTURE UNIT
COMPARATOR
PWM
PORT
69
40
INTERRUPT FLAG
INTERRUPT FLAG
INTERRUPT FLAG
EXTENSION INTERRUPT FLAG
1
(MASTER/TIMER) CORE BOARD
PCI INTERRUPT REQUEST IS PRODUCED BASED ON RESULT OF COLLECTION (OR) OF INTERRUPT FLAGS (1)
(2)
(5)
PCI INTERRUPTS ARE DISABLED
PCI INTERRUPT SIGNAL
(2)
(4)
MOTHERBOARD
PCI INTERRUPT HANDLING (3)
COMPLETION OF HANDLING
30

Fig.18B


1
RESULT OF COLLECTION (OR) OF INTERRUPT FLAGS IS TRANSMITTED TO MASTER CORE BOARD
INTERRUPT FLAG
UART/DMA (COMMUNICATION)
INTERRUPT FLAG
UART (COMMUNICATION)
INTERRUPT FLAG
PWM
INTERRUPT FLAG
AD
PORT
(SLAVE) CORE BOARD
40